ppl

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/469,277	12/22/1999	STEVEN C. DAKE	INTL-0278-US	4789

7590 08/26/2003

TIMOTHY N TROP
TROP PRUNER HU & MILES PC
8554 KATY FREEWAY STE 100
HOUSTON, TX 77024

EXAMINER	
CRAIG, DWIN M	
ART UNIT	PAPER NUMBER
2123	11

DATE MAILED: 08/26/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Paper No. 11

Application Number: 09/469,277
Filing Date: December 22, 1999
Appellant(s): DAKE ET AL.

Mark J. Rozman
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 4-30-2003.

(1) ***Real Party in Interest***

A statement identifying the real party in interest is contained in the brief.

(2) ***Related Appeals and Interferences***

The Appellant has disclosed that there are no related appeals and interferences.

(3) ***Status of Claims***

The statement of the status of the claims contained in the brief is correct.

(4) ***Status of Amendments After Final***

The amendment after final rejection filed on 1-28-2003 has not been entered.

(5) ***Summary of Invention***

The summary of invention contained in the brief is not agreed with because;

- ***Appellants refer to the specification in an attempt to show support for their invention - however; many of the indicated sections appear to be misidentified.*** For example, on page 5 of the Summary of the Invention, the Applicant identifies, in Figure 1, a set of items identified as *RAID hardware devices 16a-18a*. The Examiner observes that items *16a-18a* are actually located in Figure 4. On page 6 of the Summary of the Invention, the Applicant identifies, in Figure 10, *a disk count property 134*. The *disk count*, item is actually labeled as item 135 in the figure. Appellants do not indicate where support for the *claimed* invention resides – see 37 CFR 1.192(c)(5). Appellants have provided an exposition of the invention (pp. 2-6), which is not reflected in the claims, and has not specifically indicated where the claim limitations are supported in the specification. Furthermore, Appellants improperly introduce arguments relating to prior art techniques and alleged advantages of the present invention.

- ***Appellants have improperly introduced arguments in this section.*** This is not germane to the "Summary of the Invention". For example, *page 2, forth paragraph, "The object model API 10 may reside between the software program."* This argument is not directed towards a single embodiment illuminating the claims. Rather this argument only makes the specific limitations in the claim less clear.
- ***Appellants appear to have reinterpreted the invention.*** See, for example, in the *5th paragraph, page 2,* the appellant argues, *"The object model API may be beneficial for some hardware/software model."* There is nothing that clearly discloses how the object model may be beneficial or what the criteria is for making this determination.

(6) Issues

The appellant's statement of the issues in the brief is correct.

(7) Grouping of Claims

Appellant's brief includes a statement that claims 1-2, 6-10, 16-18, 19-21 and 31-32 for prior art rejections and 1-2, 6-10 and 19-21 are grouped for enablement rejections and provides reasons as set forth in 37 CFR 1.192(c)(7) and (c)(8).

For the purposes of appeal, the Applicants have grouped the claims according to the prior art rejections and the enablement rejections.

(8) Claims Appealed

The copy of the appealed claims contained in the Appendix to the brief is correct.

(9) Prior Art of Record

5,732,261	Savitzky et al.	3-1997
5,926,775	Brumley et al.	7-1999
6,247,077	Muller et al.	6-2001
5,877,966	Morris et al.	3-1999
5,915,253	Christiansen	6-1999

(10) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

- Claims 1-17 and 19-21 are rejected under 35 U.S.C. 112 first paragraph, as containing subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention. The specification broadly refers to modeling of hardware using object software technology. However, the specification does not provide any substantive detail to the relationships between the different software objects as disclosed. The specification does not provide any Unified Modeling Language diagrams or other figures that provide a guide as to how the different software objects interact or are dependent upon each other in order to function. The specification does not include any detail as to how the complex mechanisms for object asynchronous event notification is to be reduced to practice and in general provides no guidance to an artisan as to how to make the invention.

Claim(s) 1-17 and 19-27 are rejected under 35 U.S.C. 112, first paragraph, as containing subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention without undue experimentation.

This relates to a lack of substantive detail pertaining to the claimed features.

Claim 1, for example, recites " ***A method, comprising: defining a plurality of hardware devices as a plurality of objects;*** ". The specification does not support with enough detail the method used to define a hardware device as an object and therefore the method is not enabled. Specifically, the example on page 6, where the applicant discloses, "For example, suppose the hardware device modeled is a staple gun. A staple gun may push staples into a piece of paper. Accordingly, an object modeling the staple gun may appropriately include a method, PushStaples" along with Figure(s) 1,5 and 6 and the disclosed descriptions on page(s) 3,4,5,6 and 7 do not provide enough detail such that one of ordinary skill in the art could make and/or use the invention, without undue experimentation.

Claim 1, as another example recites, " ***providing a plurality of tools to perform a plurality of operations on the plurality of objects;*** ". The specification does not support with enough detail the method used to develop tools to perform operations on objects and is therefore not enabled. Specifically the example on page 7, " In Figure3, the tools of the configuration library **20** may include several functions. The functions may be used by the software program **12** to communicate with the objects **30**." Along with further disclosures on page(s) 8 and 9 do not provide enough detail such that one of ordinary skill in the art could make and/or use the invention, without undue experimentation.

Claim 1, as another example recites, " ***executing a software program to use the plurality of tools: and responding to the plurality of operations by the plurality of hardware devices.*** ". The specification does not support with enough detail the method used to create the software tools and the software program needed to respond to the hardware devices and is therefore not enabled. Specifically, the example on page 14 recites, " the RAID software **12a** may perform an operation to determine the devices which are located on a bus. Looking back to Figure 5, suppose the RAID software **12a** wants to determine which devices are connected to the SCSI bus **110A** of the system **14**. An operation to scan the SCSI bus **110A** ...includes invoking the method invocation function **62** of the configuration library **20** (see Figure 3). In Figure **12**, the method invocation function **62** is passed two parameters (block **202**). The object **30** for which a method **32** is invoked is provided as a parameter. ... The method **32** to invoke from the passed object is also provided as a parameter." ... from page 15 Accordingly, an integer value is stored in the bus device count memory location once the property is retrieved by the get property function **64**. The scanning operation is thus complete (**block 210**)." Along with disclosures in the figure(s) 5 and 12 are inexact and do not provide sufficient detail such that a one of ordinary skill in the art could make and/or use the invention, without undue experimentation.

Claim 2 recites, " ***assigning a plurality of properties to the plurality of hardware devices; and assigning a plurality of methods to the plurality of hardware devices.*** " The specification does not support, with sufficient detail, a clear description of how the plurality of properties and plurality of methods are assigned to the plurality of hardware devices and therefore the claim is not enabled. Specifically, the example on Figure 8 along with the description on page's 12 and 13 disclose, " the bus object includes a scan bus method **111** and several properties **34**, including a bus index property **112**, a bus identification property **113**, a bus protocol,..." which does not provide enough detailed disclosure of the specific hardware and how that hardware is modeled in software, such that one of ordinary skill in the art could make and/or use the invention, without undue experimentation.

6. Claim 3 recites, " ***assigning a plurality of events to the plurality of hardware devices.*** " The specification does not support with a sufficient level of detail a clear description of how the events are implemented and is therefore not enabled. Specifically, the description on page(s) 6-7 " ...the staple gun may run out of staples. Accordingly, an object modeling the staple gun may appropriately include an event, OutOfStaples." do not clearly define what the event is such that one of ordinary skill in the art could make and/or use the invention, without undue experimentation.

Claim 4 recites, " **providing a function for invoking a method of an object; providing a function for setting a property of an object; and providing a function for retrieving a property of an object.** " The specification does not support with a sufficient level of detail the clear definition of these function(s) and is therefore not enabled. For example, the disclosure on page 8 recites, " the configuration library **20** also provides for manipulating the properties **34** of the object **30**. A get property function **64** enables the software program **12** to retrieve the property **34** of the object **30**." As an example, this part of the specification does not clearly and exactly disclose the method used to implement the get property function such that one of ordinary skill in the art could make and/or use the invention, without undue experimentation.

Claim 5 recites, "*providing a function for monitoring an event of an object; and providing a function for ending monitoring an event of an object.*" The specification does not support with a sufficient level of detail the clear and exact definition of these function(s) and is therefore not enabled. For example, the specification on page 8 recites, " The configuration library **20** also provides functions which enable the software program **12** to be notified when an event to one of the hardware devices **16-18** occurs. A monitor event function **68**, when enabled, allows the software program **12** to be notified upon the occurrence of the particular event **36** of object **30**. Likewise, a monitor event off function **70** allows the software program **12** to turn off monitoring of the particular event **36** of the object **30**." The preceding example shows how the specification does not clearly and exactly disclose the method for creating these functions such that one of ordinary skill in the art could make and/or use the invention, without undue experimentation.

Claim 6 recites, "*An article comprising a medium storing instructions that cause a processor-based system to: receive a request from a software program,*" The specification does not support with a sufficient level of detail the clear and exact definition of how the processor based system receives a request from the software program and therefore is not enabled. For example on page 3 of the specification, " the software program may include a graphical user interface which permits a user of the software program to control and monitor hardware devices." This recitation does not disclose the exact manner that is employed to create the software program such that one of ordinary skill in the art could make and/or use the invention, without undue experimentation.

Claim 7 recites, " ***processor-based system to use a plurality of configuration library tools to act upon a plurality of objects*** " This specification does not disclose at the level of detail and clarity required such that the claim is enabled. Specifically, the specification recites, "The Configuration library **20** further includes a set of functions for object creation and object discovery." This recitation does not clearly and exactly disclose the manner in which the configuration library could be made and/or used by one of ordinary skill in the art, without undue experimentation, and is therefore not enabled.

Claim 8 recites, "***storing instructions that cause a processor based system to invoke a plurality of methods of the plurality of objects***." The specification does not support with a sufficient level of detail and in a clear and exact manner how the processor-based system invokes the methods and the objects disclosed in Claim 8 and is therefore not enabled. As an example from page 14 line 13, " the RAID software **12a** may perform an operation to determine the devices which are located on a bus" is not clear and detailed enough that one of ordinary skill in the art could make and/or use the invention, without undue experimentation.

Claim 9 recites, "*...**a processor-based system to retrieve a plurality of properties of the plurality of objects***". The specification does not provide sufficient detail in a clear and exact manner and is therefore not enabled. As an example, the specification discloses on page 13, "the RAID software **12a** may set or retrieve any of the properties or may invoke the scan bus method **111**, as desired." The specification doesn't disclose in sufficient exact detail how the objects properties are retrieved such that one of ordinary skill in the art could make and/or use the invention, without undue experimentation.

Claim 10 recites, "*... cause a processor-based system to monitor a plurality of events for the plurality of objects*" The specification does not support in sufficient detail the manner in which the system monitors the events and is therefore not enabled. For example on page 13, "...a disk is normal event **161** may permit the RAID software **12a** to be notified when the disk has been marked normal" This example, along with the rest of the specification. does not disclose, with enough detail, how the monitoring of events is accomplished such that one of ordinary skill in the art could make and/or use the invention, without undue experimentation.

Claim 11 recites, " ***A system, comprising: a processor; a plurality of hardware devices; and a medium including a software program which: models a plurality of hardware devices as a plurality of objects, wherein the plurality of objects comprise a plurality of methods and a plurality of properties; provides a plurality of tools for performing a plurality of operations on the plurality of objects; and invokes the plurality of hardware devices to respond to the plurality of operations performed on the plurality of objects.*** "The specification does not support in sufficient and clear detail the system as disclosed above and is therefore not enabled. As an example, the section of the specification on page 11 " In Figure **5**, a system **14** ...may include a processor **90** and a memory **92** are coupled to a processor bus **94**." Along with from page 12 " From the physical RAID hardware devices **17a** of Figure **5**, a plurality of RAID objects **30a** may be derived." As well as page 12 the specification recites, "...the bus object **110** includes a scan bus method **111** and several properties **34,** including a bus indexing property **112**, a bus identification property **113**, a bus protocol property **114** ...". Which does not disclose in a clear and detailed manner such that one of ordinary skill in the art could make and/or use the invention, without undue experimentation.

Claim 12 recites " ***wherein the software program further models the plurality of hardware devices as a plurality of events.*** " The specification does not support in sufficient and clear detail the software program as disclosed above and is therefore not enabled. As an example, from page 13 of the specification, "...a disk is normal event **161** may permit the RAID software **12a** to be notified when the disk has been marked normal" The monitoring of events by the software needs to be disclosed in the specification such that one of ordinary skill in the art could make and/or use the invention, without undue experimentation.

Claim 13 recites " ***...the software program further performs operations on the plurality of objects by invoking one of the plurality of methods of one of the objects.*** " The specification does not support in sufficient and clear detail the operations on the plurality of objects as disclosed above and is therefore not enabled. As an example from page 14, "An operation to scan the SCSI bus **110A** according to one embodiment of the invention includes invoking the method invocation function **62** of the configuration library **20** (see Figure **3**)" The details of how the software program performs operations on the plurality of methods of one of the objects is not clear and detailed enough such that one of ordinary skill in the art could make and/or use the invention, without undue experimentation.

Claim 14 recites, " ***...the software program further performs operations on the plurality of objects by setting one of the plurality of properties of one of the plurality of objects.*** " The specification does not support in sufficient and clear detail the operations on the plurality of objects as disclosed above and is therefore not enabled. As an example, on page 13 ""the RAID software **12a** may set or retrieve any of the properties or may invoke the scan bus method **111**, as desired." The details of how the software program sets the properties of one of the objects is not disclosed such that one of ordinary skill in the art could make and/or use the invention, without undue experimentation.

Claim 15 recites, " ***the software program further performs operations on the plurality of objects by monitoring one of the plurality of events of one of the plurality of objects.*** ". The specification does not support in sufficient and clear detail the method used to monitor the events of one of the objects and is therefore not enabled. As an example, on page 8 "The configuration library **20** also provides functions which enable the software program **12** to be notified when an event to one of the hardware devices **16-18** occurs. A monitor event function **68**, when enabled, allows the software program **12** to be notified upon the occurrence of the particular event **36** of object **30**. Likewise, a monitor event off function **70** allows the software program **12** to turn off monitoring of the particular event **36** of the object **30**." Which does not show the method for monitoring the events of the object such that one of ordinary skill in the art would be able to make and/or use the invention, without undue experimentation.

Claim 16 recites, " ***A system comprising: a processor; a plurality of disks; and a memory storing software which: models the plurality of disks as a plurality of disk objects; provides a plurality of tools for performing a plurality of operations on the plurality of disk objects; and invokes a response by the plurality of disks to the plurality of operations performed on the plurality of disk objects.*** " The specification does not support in clear detail the method for the plurality of tools to perform a plurality of operations on a plurality of disk objects and is therefore not enabled. As an example, on page 11 of the disclosure, "In Figure **5**, a system **14** ...may include a processor **90** and a memory **92** are coupled to a processor bus **94**." Along with from page 12 " From the physical RAID hardware devices **17a** of Figure 5, a plurality of RAID objects **30a** may be derived" further on page 12, " Likewise, the controller object **100** includes properties **34**, such as a bus counting method **103**, used to report the number of buses on the controller, and a disk counting method **104**, for reporting the number of disks on the controller. " This example, along with the rest of the specification, do not show enough detail such that a person of ordinary skill in the art could make and/or use the invention, without undue experimentation.

Claim 17 recites, "***The system of claim 16, wherein the software program is stored in memory.***" Which does not support in clear detail how the software program is stored in memory and is therefore not enabled. The examiner found no description in the specification whereby the software program is stored in memory. Therefore one of ordinary skill in the art could not make and/or use the invention without undue experimentation.

Claim 19 recites, "***The system of claim 18, further comprising a memory storing software which...***" which does not support in clear detail how the software program is stored in memory and is therefore not enabled. The examiner found no description in the specification whereby the software program is stored in memory. Therefore one of ordinary skill in the art could not make and/or use the invention, without undue experimentation.

Claim 20 recites, "***The system of claim 19, further comprising a memory storing software which...***" which does not support in clear detail how the software program is stored in memory and is therefore not enabled. The examiner found no description in the specification whereby the software program is stored in memory. Therefore one of ordinary skill in the art could not make and/or use the invention, without undue experimentation.

Claim 21 recites, "***The system of claim 20, further comprising a memory storing software which invokes a response to the plurality of operations by: the plurality of buses for operations performed on the plurality of bus objects; and the plurality of controllers for operations performed on the plurality of controller objects***" The specification does not support in clear detail how the software program is stored in memory and the specification does not disclose in a clear and detailed manner how the software invokes a response to a plurality of operations on the bus objects or the controller objects and is therefore not enabled. As an example on page 12 of the specification, " ...the controller object **100** includes properties **34**, such as a bus counting method **103**, used to report the number of buses on the controller, and a disk counting method **104**, for reporting the number of disks on the controller." Which does not disclose in a clear and detailed manner how the software invokes a response to operations on the controller and bus objects such that one of ordinary skill in the art could make and/or use the invention, without undue experimentation.

Claim 22 recites, "***An object comprising: a plurality of methods to model operations performed upon a device; a plurality of properties to model attributes of the device; and a plurality of events to model actions of the device.***" The specification does not disclose in a clear and detailed manner how the plurality of methods models operations, how the plurality of properties model the attribute of the device and how the plurality of events models the actions of the device and therefore is not enabled. As an example on page 5 of the specification, "The objects **30**, in essence, define the hardware devices **16-18** by modeling the operations (or methods), the attributes (or properties), and the actions (or events) of each hardware device **16-18.** An operation, known as a method, may be invoked upon an object." Which does not disclose in a clear and detailed manner the methods, properties, and events of the device objects such that one of ordinary skill in the art could make and/or use the invention, without undue experimentation.

Claim 23 recites, "...***wherein the methods comprise parameters of the object.***" The specification does not disclose with clear detail how the methods comprise the parameters of the object and therefore is not enabled. As an example, on page 8 of the specification discloses, "The method **32** may include parameters which may be set prior to invoking the method **32** or may be returned upon executing the method **32**." The specification does not disclose in a clear and detailed manner what the parameters are such that one of ordinary skill in the art could make and/or use the invention, without undue experimentation.

Claim 24 recites, "***The object of claim 23, wherein the parameters comprise properties of the object.***" The specification does not show in a clear and detailed manner how the parameters comprise the properties of the object and therefore is not enabled. As an example, on page 8 of the specification, "In one embodiment of the invention, the parameters may include properties **34** of the object **30** for which the method **32** is invoked." which does not show in a clear and detailed manner how the parameters may include the properties of the object such that one of ordinary skill in the art could make and/or use the invention, without undue experimentation.

Claim 25 recites, "***A system, comprising: an interface, comprising: a plurality of functions; and a plurality of objects coupled to the plurality of functions and a plurality of devices coupled to the interface, wherein a software program may control the plurality of devices by communicating with the interface.***" The specification does not disclose in a clear and detailed manner how the software program may control the system, functions and objects coupled to functions, and devices coupled to an interface such that the software program may control the devices via the interface and is therefore not enabled. As an example of page 11 of the specification, "In Figure 5, a system **14** ...may include a processor **90** and a memory **92** are coupled to a processor bus **94**." Along with from page 12 " From the physical RAID hardware devices **17a** of Figure **5**, a plurality of RAID objects **30a** may be derived" further on page 12, " Likewise, the controller object **100** includes properties **34,** such as a bus counting method **103,** used to report the number of buses on the controller, and a disk counting method **104,** for reporting the number of disks on the controller." And including from page 14 "...the RAID software **12a** may perform a number of operations on the RAID hardware devices **17a**. Simply by controlling the properties **34** and methods **32** of the RAID objects **30a**, the RADI software **12a** can configure the RADI hardware devices **17a**." which does not disclose the detail required for one of ordinary skill in the art to make and/or use the invention, without undue experimentation.

Claim 26 recites, "...***the plurality of functions further comprises a function for retrieving a property of one of the plurality of objects.***" The specification does not disclose in a clear and detailed manner how the function can retrieve the property from the object and is therefore not enabled. An example from page 12,"Each of the properties **34** is accessible to the

RAID software **12a** for communicating with the controller of the RAID hardware devices **17a**." which does not disclose in a clear and detailed manner how the software retrieves properties from the one of the plurality of objects such that one of ordinary skill in the art could make and/or use the invention, without undue experimentation.

Claim 27 recites, "...the function for retrieving a property of one of the plurality of functions further comprises: a parameter to identify the object for which the property is retrieved; a parameter to identify the property to be retrieved; and a parameter for storing a result." The specification does not disclose in a clear and detailed manner a parameter to identify the object, a parameter to identify the property, a parameter for storing a result and is therefore not enabled. As an example page 8 of the specification recites, "The method 32 may include parameters which may be set prior to invoking the method 32 or may be returned upon executing the method 32." And on page 9 "API 10 provides the capability to invoke the methods 32, set and retrieve the properties, 34, and monitor the events 36 for a given object 30." which does not disclose in a clear and detailed manner how a function for retrieving a property of one of the plurality of functions comprises: a parameter to identify the object for storing a result such that one of ordinary skill in the art could make and/or use the invention, without undue experimentation.

- Claims 1 & 2 are rejected under 35 U.S.C. 102(e) as being clearly anticipated by Savitsky et al. (U.S. Patent 5,732,261 – of record).

Claim(s) 1,2 are being rejected under 35 U.S.C. 102(e) as being clearly anticipated by Savitzky et al. "METHOD OF USING OBJECT-ORIENTED COMMUNICATION

SYSTEM WITH SUPPORT FOR MULTIPLE REMOTE MACHINE TYPES" U.S. Patent 5,732,261. *Savitzky et al.* Discloses, Taking claim 1, for example: A method, comprising: defining a plurality of hardware devices as a plurality of objects, From **Col. 2 Lines 66 & 67**, "the first plurality of software objects describing services for one of the plurality of remote machines" *Savitzky et al* discloses, providing a plurality of tools to perform a plurality of operations on the plurality of objects, From *Savitzky et al* **Col 2 Lines 55 & 56**, The preset invention relates generally to the area of service tools for remote machines. *Savitzky et al* discloses, executing a software program to use the plurality of tools; and responding to the plurality of operations by the plurality of hardware devices. From *Savitzky et al* **Col. 2 Lines 18 to 23**, To access a remote machine, the remote service application uses a "device driver" associated with some interface device such as a modem, and a "protocol driver" that formats the data sent to and received from the remote machine. These drivers may be part of the operating system or may be modules within the application program.

With respect to Claim 2 the *Savitzky et al* reference discloses defining the plurality of hardware devices as a plurality of objects further comprises: assigning a plurality of properties to the plurality of hardware devices; and assigning a plurality of methods to the plurality of hardware devices. From *Savitzky et al* **Col. 1, Lines 26 to 35.** As is understood to one skilled in the art. an "object" is an abstraction of a real world entity and is implemented as a combination of a data structure (whose fields are called "attributes" or "data members") and a set of operations ("methods" or "member functions") that can be performed on it. A class" is a data type for a set of objects that each have the same data structure and the same operations. An instance" of a class is an object, the data type of which is the class as actually embodied in the memory of a running application program. As recited for claims 25-27; *Savitzky et al.* discloses:

A system, comprising: an interface, comprising: a plurality of functions; and a plurality of objects coupled to the plurality of functions; and a plurality of devices coupled to the interface, wherein a software program may control the plurality of devices by communicating with the interface and wherein the plurality of functions further comprises a function for retrieving a property of one of the plurality of objects, and wherein the function for retrieving a property of one of the plurality of functions further comprises: a parameter to identify the object for which the property is retrieved; a parameter to identify the property to be retrieved; and a parameter for storing a result. As an example *Savitzky et al.* Discloses, **Col. 1 Lines 17-22** "The present invention relates generally to the area of service tools for remote machines. More specifically, the present invention relates to communication between service application programs in computer systems, and remote devices such as remote machines and external data in files, databases, and programs. " As another example *Savitzky et al.* Discloses, **Col. 1 Lines 24-35** "An understanding of object oriented programming and object-oriented application frameworks will assist in full understanding of the present invention. As is understood to one skilled in the art, an "object" is an abstraction of a real world entity and is implemented as a combination of a data structure (whose fields are called "attributes" or "data members") and a set of operations ("methods" or "member functions") that can be performed on it. A "class" is a data type for a set of objects that each have the same data structure and the same operations. An "instance" of a class is an object, the data type of which is the class, as actually embodied in the memory of a running application program. "

- Claims 3-5 are rejected under 35 U.S.C. 103(a) as being unpatentable over Savitsky et al. (U.S. Patent 5,732,261 – of record) in view of Brumley et al. (U.S. Patent 5,926,775 – of record).

Claim(s) 3, 4 and 5 are being rejected under 35 U.S.C. 103(a) as being unpatentable over **Savitzky et al**. in view of **Brumley et al**. **U.S. Patent 5,926,775** As noted above *Savitzky et al.* discloses A method, comprising: defining a plurality of hardware devices as a plurality of objects: Providing a plurality of tools to perform a plurality of operations on the plurality of objects; Executing a software program to use the plurality of tools and responding to the plurality of operations by the plurality of hardware devices; assigning a plurality of properties to the plurality of hardware devices; and assigning a plurality of methods to the plurality of hardware devices. However the *Savitzky et al.* does not expressly disclose assigning a plurality of events to the plurality of hardware devices. The *Brumley et al.* reference discloses in **Col. 12 Lines 46-49.** "The Message Manager handles messaging to the user application in response to events or interrupts." It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify *Savitzky et al.* with *Brumley et al.* and assign a plurality of events to the plurality of hardware devices because the mechanism of reporting driver status via event handling is very efficient as opposed to other methods of obtaining current status.

With respect to Claim 4 *Brumley et al.* doesn't expressly disclose a function for invoking a method of an object. However, *Savitzky et al.* recites, **Col. 16, Lines 28-31,** "if the SI.sub.--AppItem is marked "busy" a callback method is invoked when the work is actually done." It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify *Savitzky et al.* with *Brumley et al.* to provide a function for invoking a method of an object because otherwise the object could not be used.

With respect to Claim 4 both *Brumley et al.* and *Savitzky et al* teach a function for setting a property of an object and a function for retrieving a property of an object. For example *Savitzky et al.* recites, **Col. 4 Lines 60-63.** "If there is a member function that updates a value, its name ends in "Set"; the update function is declared by the same macro that declares the attribute." Also *Brumley et al.* recites **Col. 23 Lines 61-64.** Once reserved, Get(attrID, attrVal) and Set(attrID, attrVal) attribute calls provide the interface for setting up and querying current programmable attributes. It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify *Savitzky et al* with *Brumley et al.* to provide function(s) for setting and retrieving a property of an object because that is the manner in which attributes are accessed and modified in object oriented programming.

- Claims 6-15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Brumley et al. (U.S. Patent 5,926,775 – of record) in view of Muller et al. (U.S. Patent 6,247,077 – of record).

Claim(s) 6 - 15 are being rejected under 35 U.S.C. 103(a) as being unpatentable over **Brumlet et al. U.S. Patent 5,926,775** in view of **Muller et al. U.S. Patent 6,247,077.**

The *Brumlet et al.* discloses, as recited from claims 6-10; "An article comprising a medium storing instructions that cause a processor-based system to: receive a request from a software program; act upon a plurality of objects based upon the request received; and manipulate a plurality of hardware devices modeled by the plurality of objects." **Figure(s) 2 thru 17 and Col. 27 Lines 6 thru 33.**

As regards Applicants amended **Claim 6,** the *Brumlet et al.* reference does not expressly disclose a, "redundant array of independent disks modeled by the plurality of objects".

The *Muller et al.* reference discloses a redundant array of independent disks modeled by the plurality of objects, **Figure 1 element 104, Figure 2 element 218 and 222, Figure 3 elements 222 and 218, Figure 4 elements 402 thru 416, Figure 5 elements 502, 504, 516, 500, 216, 224 and 222 and Col. 5 Lines 18-67 and Col.6 Lines 1-67 and Col. 7 Lines 1-35 and Col. 8 lines 1-38 as well as the Abstract, Background of the Invention and Detailed Description of the Preferred Embodiment.**

It would have been obvious to one of ordinary skill in the art, at the time of the invention, to have modified the *Brumlet et al.* reference with the *Muller et al.* reference because, *(motivation to combine)* "the uneven technology growth...requires a fundamentally different storage connectivity model-one which allows workload scaling to match technology improvements, **Muller et al. Col. 3 Lines 22-26.**

As regards to Claim 7 **Col. 4 Lines 18-24.** *Brumlet et al.* Recites, "A driver family interpreter executes on top of the mini-driver primitives. The interpreter operates to configure device family independent or hardware independent features for each device. The interpreter can also operate to convert from a legacy user interface to the low level programming interface provided by the respective mini-driver primitives." As recited from claims 11-15, *Brumlet et al.* discloses: A system comprising: a processor; **Col. 6 Lines 44 thru 49.** a plurality of hardware devices, **Col. 7 Lines 7 thru 13.** a medium including a software program which: models the plurality of hardware devices as a plurality of objects, wherein the plurality of objects comprise a plurality of methods and a plurality of properties; **Figure(s) 2,3,4,7,11,12,13** and **Col. 3 Lines 21-28** "The DAQ driver level software also includes a plurality of mini-driver primitives. Each of the mini-driver primitives performs a portion of controlling the DAQ device. In other words, the DAQ device includes a plurality of hardware resources for performing DAQ functions, and each of the plurality of mini-driver primitives controls one of the hardware resources." and *Brumlet et al.* Recites, **Col. 13 Lines 26-30** "Primitives have a set of attributes, which themselves have values. Primitives also support a set of control codes, which are used in conjunction with the Control method on a primitive."

With respect to Claim 15. The *Brumley et al.* reference discloses in **Col. 12 Lines 46-49.** "The Message Manager handles messaging to the user application in response to events or interrupts."

Furthermore, *Brumley et al.* discloses: An object comprising: a plurality of methods to model operations performed upon a device; a plurality of properties to model attributes of the device; and a plurality of events to model actions of the device, wherein the methods comprise parameters of the object and wherein the parameters comprise properties of the object. *Brumley et al.* Recites **Figure(s) 2 & 3** and **Col. 23 Lines 61-64** "Once reserved, Get(attrID, attrVal) and Set(attrID, attrVal) attribute calls provide the interface for setting up and querying current programmable attributes" As regards to "*Events*" **Col. 12 Lines 46-49.** "The Message Manager handles messaging to the user application in response to events or interrupts."

- Claims 16-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Brumley et al. (U.S. Patent 5,926,775 – of record) in view of Morris et al. (U.S. Patent 5,877,966 – of record).

Claim(s) 16 - 21 are being rejected under 35 U.S.C. 103(a) as being unpatentable over **Brumley et al. U.S. Patent 5,926,775** in view of **Morris et al. U.S. Patent 5,877,966** *Brumley et al.* discloses A system, comprising: a processor; a plurality of disks; and memory storing software. Also *Brumley et al.* recites **Col. 23 Lines 61-64.** Once reserved, Get(attrID, attrVal) and Set(attrID, attrVal) attribute calls provide the interface for setting up and querying current programmable attributes. It would have been obvious to one of ordinary skill in the art at the time the invention was made to provide function(s) for setting and retrieving a property of an object. Which can be restated to being, "plurality of tools for performing a plurality of operations on "some type" of object(s)". *Brumley et al.* teaches multiple buses and multiple controllers. **Figure 1, Figure 1A.** and further teaches multiple bus objects and multiple controller objects. **Figure(s) 11,12 and 13.**

However, *Brumley et al.* does not expressly disclose is: models the plurality of disks as a plurality of disk objects. *Morris et al.* teaches, **Figure 2 and Col. 5 Lines 54 – 57.** "These constituent parts are, in turn, members of a "class" of objects having similar features. For example, the 300 MB SCSI Hard Disk part 58 is a member of the class "300 MD SCSI Hard Disk" 66. Class 66 is, in turn, a subclass of class "Generic Hard Disk" 72, as are subclasses 64, 68 and 70." *Morris et la.* also teaches a plurality of volumes and a plurality of arrays. **Col. 5 Lines 29-39.** At the time of the invention it would have been obvious to one of ordinary skill in the art to modify *Brumley* with *Morris* in order to represent disk drives as a plurality of disk objects and be able to perform a plurality of operations on said disk objects, since the objective is to model the computer system along with its data storage sub-systems using an object oriented paradigm.

- Claims 28-32 are rejected under 35 U.S.C. 103(a) as being unpatentable over Savitzky et al. (U.S. Patent 5,732,261 – of record) in view of Muller et al. (U.S. Patent 6,247,077 – of record), and in further view of Christiansen (U.S. Patent 5,915,253).

Claim(s) 28-32 are being rejected under 35 U.S.C. 103(a) as being unpatentable over **Savitzky et al. U.S. Patent 5,732,261** in view of **Muller et al. U.S. Patent 6,247,077** and in further view of **Christiansen U.S. Patent 5,915,253.**

As regards to Applicants new **Claims 28-32,** the limitations of **Claim 1** have been rejected as being anticipated by the *Savitzky et al.* reference, see Examiners rejection paragraph 31 above.

As regards **Claims 28 and 29** the *Savitzky et al.* reference does not expressly teach does not expressly disclose a, "redundant array of independent disks modeled by the plurality of objects".

The *Muller et al.* reference discloses a redundant array of independent disks modeled by the plurality of objects, **Figure 1 element 104, Figure 2 element 218 and 222, Figure 3 elements 222 and 218, Figure 4 elements 402 thru 416, Figure 5 elements 502, 504, 516, 500, 216, 224 and 222 and Col. 5 Lines 18-67 and Col.6 Lines 1-67 and Col. 7 Lines 1-35 and Col. 8 lines 1-38 as well as the Abstract, Background of the Invention and Detailed Description of the Preferred Embodiment.**

It would have been obvious to one of ordinary skill in the art, at the time of the invention, to have modified the *Savitzky et al.* reference with the *Muller et al.* reference because *(motivation to combine)* , "the uneven technology growth...requires a fundamentally different storage connectivity model-one which allows workload scaling to match technology improvements, **Muller et al. Col. 3 Lines 22-26.**

As regards **Claims 30** the *Savitzky et al.* reference does not expressly disclose a Volume Object or Disk Objects.

The *Muller et al.* reference discloses a redundant array of independent disks modeled by the plurality of objects, *see 35 U.S.C.103 rejection directly above in this paragraph.*

The *Christiansen* reference discloses a Volume Object **Figure 6 Item 180, and Figure 7 Item 180**.

It would have been obvious to one of ordinary skill in the art, at the time of the invention, to have combined the *Savitzky et al.* reference with the *Christiansen* reference because *(Motivation to combine)* modularization and abstraction of the a computer systems file system allows for greater flexibility ***(Christiansen Col. 1 Lines 1-67 and Col. 2 Lines 1-67 and Col. 3 Lines 1-31).***

As regards **Claims 31-32** the *Savitzky et al.* reference teaches at least one method and at least one property and at least one event as well as communicating using at least one property *(see Examiners rejection of Claim 1 above).*

(11) Response to Arguments

A. As regards Appellant's arguments regarding the rejection of Claims 1& 2 under 35 U.S.C. 102(e) as being clearly anticipated by Savitsky et al. (U.S. Patent 5,732,261 – of record).

The *Savitsky et al.* reference discloses a method of communicating with a plurality of remote machines (hardware) using a computer system, including the steps of constructing in memory software objects that describe the services of the remote machines, establishing communications with the remote machines and invoking operations on the remote machines in response to requests described by services of the plurality of the software objects.

The Appellants have argued that the *Savitsky et al.* reference does not disclose "defining a plurality of hardware devices as a plurality of objects." *see page 8, paper#10.* The Examiner asserts that in Figure 2 of the *Savitsky et al.* reference is a diagram of hardware devices (elements 200 and 210) being defined as a plurality of software objects (Item 160), the Examiner notes that "Base Classes, Item 160" are the software objects that are used to define the Hardware Devices (Item 200 Copier and Item 210 Fax).

Examiner asserts that the *Savitsky* reference does disclose defining a plurality of hardware devices as a plurality of objects, a ***Software object*** is the same as ***an object***, a ***remote machine*** in the context of the *Savitsky* reference is a **hardware device**, examiner asserts that in **Col. 3 Lines 60 to 67** the *Savitsky* reference discloses in greater detail, **[A "component" is a software object that represents the services and state of some**

remote machine. A component may have sub-components, e.g. a copier may have a sorter and a feeder attached to it. A "device" is a computer system peripheral that enables the computer system in which the remote service application is running to communicate with one or more remote machines.], and further in **Col. 4 Lines 1-4** *Savitsky* discloses, **[A "device driver" is a software object that provides an interface by which the remote service application communicates with a device.]** to summarize, the *Savitsky* reference discloses defining a plurality of hardware devices, *remote fax machines computer system* ***peripheral*** *sub-components...sorter, feeder,* represented as a plurality of objects, *A " component" is a software* ***object*** *that represents the services and state of the remote* ***machine****...A "device driver" is a software* ***object****.*

Further, the *Savitsky et al.* reference discloses providing a plurality of tools to perform a plurality of operations on the plurality of objects, from *Savitzky et al* **Col 2 Lines 55 & 56**, "The present invention relates generally to the area of service tools for remote machines." *Savitzky et al* discloses, executing a software program to use the plurality of tools; and responding to the plurality of operations by the plurality of hardware devices, from *Savitzky et al* **Col. 2 Lines 18 to 23**, to access a remote machine, the remote service application uses a "device driver" associated with some interface device such as a modem, and a "protocol driver", that formats the data, sent to and received from, the remote machine. These drivers may be part of the operating system or may be modules within the application program. The Examiner asserts that the *Savitzky et al.* reference discloses all of the limitations in Independent Claim 1 and dependent Claim 2.

B. As regards Appellants arguments that **Claims 6-10** are patentable over the Brumley et al. (U.S. Patent 5,926,775 – of record) in view of Muller et al. (U.S. Patent 6,247,077 – of record).

The Appellant has asserted that neither the *Brunley et al.* reference or the *Muller et al.* reference disclose, *" a redundant array of independent disks modeled by the plurality a objects."* The Appellant has asserted that Claim 6 recites, that the redundant array of independent disks are an array of RAID disks *(see page 8, paper#10).* The Examiner notes that the Appellant is not using any *(means for)* language in Claim 6 and therefore cannot read into the claim language any limitations from the specification.

- In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., RAID) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

The *Muller et al.* reference discloses a redundant array of independent disks modeled by the plurality of objects, Figure 1 element 104, Figure 2 element 218 and 222, Figure 3 elements 222 and 218, Figure 4 elements 402 thru 416, Figure 5 elements 502, 504, 516, 500, 216, 224 and 222 and Col. 5 Lines 18-67 and Col.6 Lines 1-67 and Col. 7 Lines 1-35 and Col. 8 lines 1-38 as well as the Abstract, Background of the Invention and Detailed Description of the Preferred Embodiment.

The Examiner assets that the, *redundant array* as argued by the Appellant, is disclosed in the *Muller et al.* reference **(Col. 5 Lines 63-67, Col. 6 Lines 1-4, Col. 11 Lines 10-22),** specifically, a **RAID (Col. 5 Lines 63-67, Col. 6 Lines 1-4, Col. 11 Line 11)** array is disclosed.

The Examiner asserts that the *Muller et al.* reference discloses the *redundant array* as plurality of objects **(Col. 17 Lines 8-56)**. Figure 2 discloses how the **(ION or I/O Nodes)** abstract the array's of redundant disks **(JBOD or "just a bunch of disks" Col. 4 Lines 58-64).**

The Examiner asserts that the motivation to combine the *Muller et al.* reference with the *Brunley et al.* reference can be found in the *Muller et al.* reference which states that the existing difference between the rate at which processor speed is out stripping computer storage speed creates *serious problems* due to uneven growth in hardware technologies **(Col. 2 Lines 47-67, Col. 3 Lines 1-4)** and that the *Muller et al.* reference presents a powerful solution to compensate for the *serious problems* **(Col. 3 Lines 26-45)**. Specifically, the *Muller et al.* discloses that by abstracting a large collection of redundant arrays of storage media, using software drivers, *(that are object abstractions of the storage media)* the problems of not being able to realize the performance gain not being realized are overcome.

C. As regards Appellants arguments that **Claims 16-18** are patentable over Brumley et al. (U.S. Patent 5,926,775 – of record) in view of Morris et al. (U.S. Patent 5,877,966).

The Appellant has argued that there is no motivation to combine the *Brumley et al.* reference with the *Morris et al.* reference. The Examiner asserts that the same flexibility of design and the reduction of complexity that the *Morris et al.* reference discloses, **(Col. 1 Lines 51-59, Col. 3 Lines 34-51),** by using software object abstraction of computer hardware, *as* *disclosed in the Morris et al.* reference **(All of Figure 4),** would motivate an artisan of ordinary skill, at the time of the invention, to use the disclosed object technology, that the *Morris et al.* reference discloses, **(Figure 5),** to model a disk array and a complete computer system.

The abstraction of the detail functioning of the different components in a computer system provides a simple method of having those components interact, without a programmer having to understand all of the complexities of the lower level details of implementation. The idea of abstracting a software module, using object technology, to reduce the complexity is one of the primary motivations for using software abstraction. An example of the concept of software abstraction of hardware is the development of device drivers. Before operating systems had device drivers, programmers had to write all of the code required to operate a new piece of hardware, specifically, interrupt service routines, memory allocation, and error/fault condition handling. With the development of device drivers, hardware resources in computers were abstracted with software objects, *i.e. device drivers* and the software implementation details of how hardware devices functioned were abstracted in such a manner that the applications programmer used a predefined software interface to the device driver and was relieved of the burden of all of the lower implementation details.

Therefore the Examiner asserts that there is ample motivation to combine the *Brumley et al.* reference with the *Morris et al.* reference.

As regards the Appellants argument that; *...Nowhere does Morris teach or suggest software which "models the plurality of disks as a plurality of disk objects" nor "provides a plurality of tools for performing a plurality of operations on a plurality of disk objects" nor "invokes a response to by the plurality of disks to the plurality of operations performed on the* *plurality of disk objects"*. The Examiner asserts that, as per the original rejection, the *Morris et al.* reference discloses modeling a plurality of disks as disk objects **(Figure 4)**, and the Examiner relies on the *Brumley et al.* reference to reject the *plurality of tools limitations as disclosed,* ***(Brumley et al. Figure 2, DAQ Application, DAQ Driver Level Software, these are a plurality of tools, and Figure 7, Col. 12 Lines 19-43).*** The *plurality of tools perform a plurality of operations on the plurality of objects* ***(Brumley et al. Col. 12 Lines 49-65), and invokes a response (Col. 12 Lines 44-48).***

The Examiner asserts that a *prima fascia* case of obviousness has been established for combining the *Brumley et al.* reference with the *Morris et al.* reference in that when abstracting hardware devices with software objects that certain methodologies are used *(as demonstrated by the Brumley et al. reference),* and that the idea of modeling disk storage as a software object is known in the art *(Morris et al.)*.

D. As regards Appellants arguments that **Claims 6-10** are patentable over the Brumley et al. (U.S. Patent 5,926,775 – of record) in view of Morris et al. (U.S. Patent 5,877,966 – of record).

Appellant has argued that neither the *Brumley et al.* reference nor the *Morris et al.* reference disclose a *modeling a plurality of buses as a plurality of bus objects nor modeling a plurality of controllers as a plurality of controller objects.* The Examiner asserts that the *Brumley et al.* reference discloses a plurality of buses modeled as a plurality of bus objects, **(Figure 14, Items labeled "BUS OBJECT").** The Examiner asserts that the *Brumley et al.* reference discloses a plurality of controllers as a plurality of controller objects, the Examiner asserts that the **DAQ** level driver disclosed in the *Brumley et al.* reference discloses a controller object in that the **DAQ** hardware **(Col. 3 Lines 22-52, Figure 1 ITEMS 104 and ITEMS 124 which are a plurality of controllers),** is what is controlling the interface to the **Bus** objects to the devices on the bus **(Figure 2 "DAQ Driver Level Software" and Figure 3 "MINI DRIVER PRIMITIVES" under the "DAQ DEVICE OBJECT" are the controller objects and Figure 4 ITEM 204 & 208.)** The Examiner asserts that the *Brumley et al.* reference discloses a plurality of buses modeled as a plurality of bus objects and a plurality of controllers modeled as a plurality of controller objects.

E. As regards Appellants arguments that **Claim 28** is patentable over Savitsky et al. (U.S. Patent 5,732,261 – of record) in view of Muller et al. (U.S. Patent 6,247,077 – of record).

The Appellant has argued that the *Muller et al.* reference does not teach or suggest defining a redundant array of independent disks as a plurality of objects.

The Examiner asserts, as discussed above, *see section B*, the *Muller et al.* discloses a redundant array of independent disks defined as a plurality of objects.

The Examiner assets that the, *redundant array* as argued by the Appellant, is disclosed in the *Muller et al.* reference **(Col. 5 Lines 63-67, Col. 6 Lines 1-4, Col. 11 Lines 10-22),** specifically, a **RAID (Col. 5 Lines 63-67, Col. 6 Lines 1-4, Col. 11 Line 11)** array is disclosed.

The Examiner asserts that the *Muller et al.* reference discloses the *redundant array* as plurality of objects **(Col. 17 Lines 8-56)**. Figure 2 discloses how the **(ION or I/O Nodes)** abstract the array's of redundant disks **(JBOD or "just a bunch of disks" Col. 4 Lines 58-64).**

F. As regards Appellants arguments that **Claims 29 and 31-32** are patentable over Savitsky et al. (U.S. Patent 5,732,261 – of record) in view of Muller et al. (U.S. Patent 6,247,077 – of record).

The Appellant has argued that the *Muller et al.* or the *Savitsky et al.* reference does not teach or suggest defining a redundant array of independent disks as a plurality of objects.

The Examiner asserts, as discussed above, *see section B*, the *Muller et al.* discloses a redundant array of independent disks defined as a plurality of objects.

The Examiner assets that the, *redundant array* as argued by the Appellant, is disclosed in the *Muller et al.* reference **(Col. 5 Lines 63-67, Col. 6 Lines 1-4, Col. 11 Lines 10-22),** specifically, a **RAID (Col. 5 Lines 63-67, Col. 6 Lines 1-4, Col. 11 Line 11)** array is disclosed.

The Examiner asserts that the *Muller et al.* reference discloses the *redundant array* as plurality of objects **(Col. 17 Lines 8-56)**. Figure 2 discloses how the **(ION or I/O Nodes)** abstract the array's of redundant disks **(JBOD or "just a bunch of disks" Col. 4 Lines 58-64).**

G. As regards Appellants arguments that **Claim 30** is patentable over Savitsky et al. (U.S. Patent 5,732,261 – of record) in view of Muller et al. (U.S. Patent 6,247,077 – of record) and in further view of Christiansen (U.S. Patent 5,915,253 – of record).

The Examiner asserts, as stated previously, that the *Muller et al.* reference discloses *see section B*, a redundant array of disk objects.

The Examiner assets that the, *redundant array* as argued by the Appellant, is disclosed in the *Muller et al.* reference **(Col. 5 Lines 63-67, Col. 6 Lines 1-4, Col. 11 Lines 10-22)**, specifically, a **RAID (Col. 5 Lines 63-67, Col. 6 Lines 1-4, Col. 11 Line 11)** array is disclosed.

The Examiner asserts that the *Muller et al.* reference discloses the *redundant array* as plurality of objects **(Col. 17 Lines 8-56)**. Figure 2 discloses how the **(ION or I/O Nodes)** abstract the array's of redundant disks **(JBOD or "just a bunch of disks" Col. 4 Lines 58-64).**

The Examiner assets that the *Christiansen* reference does suggest, an *array of storage objects*, **(Figure 1, STORAGE SYSTEM, ITEMS 28, 30 and 34, Col. 2 Lines 11-15)** and the *Christiansen* reference discloses a volume object **(Figure 6, ITEM 180).**

H. thru K. Regarding Appellants arguments that Claims 1-17 and 19-21 are enabled under 35 U.S.C. 112 first paragraph.

Appellants have argued that the Examiner misperceives the standard for enablement.

The MPEP states:

2164.01(b) How to Make the Claimed Invention
As long as the specification discloses at least one method for making and using the claimed invention that bears a reasonable correlation to the entire scope of the claim, then the enablement requirement of 35 U.S.C. 112 is satisfied. In re Fisher, 427 F.2d 833, 839, 166 USPQ 18, 24 (CCPA 1970). Failure to disclose other methods by which the claimed invention may be made does not render a claim invalid under 35 U.S.C. 112. Spectra-Physics, Inc. v. Coherent, Inc., 827 F.2d 1524, 1533, 3 USPQ2d 1737, 1743 (Fed. Cir.), cert. denied, 484 U.S. 954 (1987).

The Examiner asserts that nowhere in Appellant's specification is there a description on how to "Make" the invention.

The Examiner asserts that if the Appellant had disclosed a UML diagram then there would be a schematic available to one of ordinary skill in the art to *make* appellant's invention.

For example, in the *Brumley et al.* reference, "*Figures 11-13 are good examples of properly disclosed object class definitions and how each of the different objects are related to each other, note the term BUS OBJECT with the diamond symbol and the words IS COMPOSED OF showing the relationship between the different classes and sub-classes.*" Appellant's specification provides no guidance or figures or diagrams to instruct an artisan of ordinary skill in the art, what the relationship between the different objects is to one another. The Examiner asserts that on of ordinary skill in the art would have to perform a long design phase, to conceptually determine which classes where base classes, if one class would *encapsulate* one of the other classes, which classes would have to be sub-classes.

As an example it is unclear to the Examiner how the Disk Object *Figure 9,* the Array Object *Figure 10* and the Volume Object *Figure 11,* are related. The Examiner asserts that it would make sense that a Disk Object would have a Volume Object *encapsulated* into it and that the *Array* object would have multiple *Disk Objects* encapsulated into it but, Appellant's specification never defines that relationship using a lexicon that an artisan of ordinary skill in the art would understand.

In regards to Appellant's response to the Claims 6-10 being rejected under 35 U.S.C. 112 as lacking enablement the Appellant is over simplifying and mischaracterizing all of the rejections as stated in the Office Action. Appellant is using only one example claim rejection and has failed to make a prima fascia case using vague allegations without addressing the arguments from the last office action. Appellant is required to show *factual* evidence of support in the specification as to how the invention would be made. The Examiner in the last office action clearly stated that:

Examiner asserts that Applicants specification fails to properly enable such that one of ordinary skill in the art could make and or use the invention without undue experimentation, **as stated in the rejection.** *Examiner further asserts that for one of ordinary skill in the art to enable Applicant's invention, the one of ordinary skill in the art would require a "wrapper", which is well known in the art, to provide a level of abstraction to facilitate porting existing code between different operating systems with out the requirement of coding a specific implementation for different computer operating systems. The Examiner further asserts that Applicant's specification does not contain the interface to the objects that are modeling or abstracting the hardware. The Examiner further asserts, that nowhere in applicant's specification, is there a mention of the relationship between the different objects, i.e. does the controller object inherit the bus object, does the array object encapsulate the volume object, which of the Applicant's objects are base classes? The Examiner further asserts that the Applicant's specification does not provide a Unified Modeling Language, (UML) diagram to represent the manner in which the Applicants objects interact and relate to each other. The Examiner further asserts that there are no source code examples or function prototypes, data structures or other implementation examples that would support that the invention has been reduced to practice. The Examiner asserts that by "Applicants own admission" see* ***page 4, lines 16-17*** *of applicants specification, the Applicant states, "The hardware device* ***16*** *may itself include software, such as a software interface* ***<u>(not shown)</u>****", this type of software interface needs to be shown to support enablement of the specification, the Examiner asserts that it would require to much experimentation on the part of one of ordinary skill in the art to have to guess how the software interface that is* ***<u>not shown</u>*** *functions.*

The Applicant has asserted, *that sending requests from a software program to a processor-based system is clearly well-known and need not be described in detail*, its is noted that the lack of details on how to *make* Appellant's invention is the reason the Examiner rejected the claims for lacking enablement.

As stated, the Examiner in the last office action *(paper # 5)*, provided factors as to why Appellant's "well-known" processes are not enabled.

As regards Appellant's response to the rejections of **Claim 16** under 35 U.S.C. 112 for lacking enablement, please see the previous discussion.

As regards Appellant's response to the rejections of **Claim 17 and 19-21** under 35 U.S.C. 112 for lacking enablement, please see the previous discussion.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,
Dwin M. Craig

Dwin McTaggart Craig
August 25, 2003

Conferees
William Thomson
Kevin Teska

KEVIN J. TESKA
SUPERVISORY
PATENT EXAMINER

TIMOTHY N TROP
TROP PRUNER HU & MILES PC
8554 KATY FREEWAY STE 100
HOUSTON, TX 77024